

## Amendments to the Specification

Please replace the title on page 1 with the following amended title:

### SYSTEM AND COMPUTER-IMPLEMENTED METHOD AND SYSTEM FOR CREATING CONFIGURATION FILES FOR WEB TRANSACTION TESTS

Please replace paragraph [0002] with the following amended paragraph:

[0002] This invention relates generally to wide area networks, and more particularly to systems and methods for ~~creating test~~ creating tests to evaluate web sites hosted by web servers coupled ~~to a TCP/IP~~ to a Transmission Control Protocol/Internet Protocol (TCP/IP) protocol network.

Please replace paragraph [0004] with the following amended paragraph:

[0004] It has become increasingly easy for companies and individuals to create their own web sites. There are a number of commercially available software applications which generate the necessary ~~HTML code~~ Hypertext Markup Language (HTML) code to create "web pages" which can be uploaded to web servers connected to the Internet. There are also web sites that can be used to create content for other web sites. Connections to the Internet and web server hosts are often provided by a company known as an Internet Service Provider (ISP).

Please replace paragraph [0012] with the following amended paragraph:

[0012] A method for creating a web transaction test in accordance with the present invention sends formatted information comprising a parsed testing form to a web site, preferably through a web server. The testing form may be parsed in anticipation of being sent, or may be parsed once received. At least one section of the parsed test form contains a part to be filled out from a predetermined list, and preferably the predetermined list is based on information contained in the web site. Also in a preferred embodiment, subsequent sections to be filled out may have predetermined lists dependent upon prior selections. In all such embodiments, in the next step a formatted message is received from the web site as a

submission from the testing form, and a test configuration file is developed from the received testing form. The formatted information and messages may be HTML, ~~HTTP~~, or Hypertext Transfer Protocol (HTTP), or other standards involving web sites. The method may also create a web transaction test from an existing test configuration file by developing a filled testing form from an existing test configuration file, sending HTML comprising the filled testing form to a web site, receiving HTTP from the web site as a submission from the testing form, and updating the existing test configuration file from the received HTTP.

Please replace paragraph [0014] with the following amended paragraph:

[0014] In a preferred embodiment, the method is capable of direct HTTP and HTML communication with the web site to be tested over a TCP/IP protocol network. Thus, the method does not require a web site to test or monitor a web site. The test configuration file may provide tests for several dynamic aspects of a web site ~~such as a URL~~ such as a Uniform Resource Locator (URL), a link, a form button, and a frame. Further, the test configuration file may provide data to post in a field parameter of the web site. Errors in the web site can be detected by comparing formatted information received from the web server with predetermined data that may be included in the test configuration file. Automatic scheduling information, including window information and test repetition information, may also be contained in the test configuration file.

Please replace paragraph [0015] with the following amended paragraph:

[0015] A system for creating a web site transaction test in accordance with the present invention includes a TCP/IP protocol network, a web server, and a computer. The web server is connected to the TCP/IP protocol network and hosts a web site. The computer has memory and a processor and is also connected to the TCP/IP protocol network. The computer sends formatted information such as an HTML document comprising a parsed testing form to a web site. As described in the method embodiment above, the parsing of the test form is preferably based on information contained on the web server, and various selection choices may also be dynamically created based on that or other information. Next, the computer receives a formatted message such as HTTP from the web site as a

submission from the testing form. The computer then develops a test configuration file from the formatted message received from the web site. The computer may also create or modify a test from an already existing test configuration file. In such instances, the computer fills a testing form from a test configuration file, sends HTML with the filled testing form in HTML format to a web site, receives HTTP form ~~the web site~~ from the web site as a submission from the testing form, and updates the test configuration file with the HTTP form received from the web browser.

Please replace paragraph [0018] with the following amended paragraph:

[0018] The invention, together with further advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

Figure 1 is an illustration of a system including the Internet, a number of client machines, a web server, and a testing computer;

Figure 2 is a pictorial representation of a testing computer 22' of the present invention;

Figure 2A is a pictorial representation of testing software implementing the present invention, as well as standard web browser software;

Figure 2B is a flow-diagram illustrating the overall operation of the present invention;

Figure 3A, Figure 3B, Figure 3C, Figure 3D, and Figure 3E illustrate a series of HTML web pages that will be used as an example when explaining the present invention;

Figure 4A, Figure 4B, and Figure 4C illustrate a blank form used to develop the testing configuration file of the present invention;

Figure 5 is ~~a illustration~~ an illustration which helps ~~an explain~~ explain a HTML "form";

Figure 6 is an exemplary test configuration developed by the process of the present invention from the web pages of Figs. 3A-3E;

Figure 7 is a flow-diagram illustrating the "ANALYZE CONFIGURATION FILE ACTIVATION SCHEDULE" operation of Fig. 2B;

Figure 8 is a flow-diagram of the "SEND HTTP TO WEB SITE AND RECEIVE HTTP/HTML" operation of Fig. 2B;

Figure 8A is a flow-diagram detailing “FIND ALL FIELDS ON FORM WITH SUBMIT BUTTON” operation of Fig. 8;

Figure 8B is a flow-diagram illustrating the “TAKE SUPPLIED POST DATA AND OVERWRITE VALUES” operation of Fig. 8;

Figure 9 is a flow-diagram illustrating the “ANALYZE AND STORE RESULTS IN LOG FILE” operation of Fig. 2B;

Figure 10 is a flowchart showing a method for creating a monitor in an automated manner, in accordance with an embodiment of the present invention;

Figure 10A is a flowchart showing a parsing process for parsing an HTML page for tags, in accordance with an embodiment of the present invention;

Figure 10B is an illustration showing a step2 monitor editing page, in accordance with an embodiment of the present invention;

Figure 10C is an illustration showing a step3 monitor editing page, in accordance with an embodiment of the present invention;

Figure 10D is an illustration showing a step4 monitor editing page, in accordance with an embodiment of the present invention;

~~Figure 10e~~ Figure 10E is an illustration showing a step5 monitor editing page, in accordance with an embodiment of the present invention;

~~Figure 10f~~ Figure 10F is an illustration showing a step6 monitor editing page, in accordance with an embodiment of the present invention;

~~Figure 10g~~ Figure 10G is an illustration showing a completed monitor display, in accordance with an embodiment of the present invention;

Figure 11 is an illustration showing an exemplary hardware configuration for a monitor system, in accordance with an embodiment of the present invention;

Figure 12 is an illustration showing an Enterprise portal system, in accordance with an embodiment of the present invention;

~~Figure 13 is dataflow~~ is a dataflow diagram showing an Enterprise portal system, in accordance with another embodiment of the present invention;

Figure 14 is a flowchart showing an ebusiness transaction chain monitor process, in accordance with an embodiment of the present invention;

Figure 15 is an illustration showing a hardware configuration suitable for executing the ebusiness transaction chain process, in accordance with an embodiment of the present invention;

Figure 16A is an illustration showing an ebusiness transaction chain monitor interface, in accordance with an embodiment of the present invention;

Figure 16B is an illustration showing an ebusiness transaction chain monitor interface ~~interface~~, in accordance with another embodiment of the present invention;

Figure 17A is a flowchart showing an email conformation process, in accordance with an embodiment of the present invention;

Figure 17B is flowchart showing a database confirmation process, in accordance with one aspect of the present invention; and

Figure 17C is a flowchart showing a script checking process, in accordance with another aspect of the present invention.

Figure 18 is an illustration of a monitoring system, in accordance with an embodiment of the present invention;

Figure 19 is a flow-diagram showing a business method for providing a monitoring system, in accordance with one aspect of the present invention; and

Figure 20 is a flow-diagram illustrating the "PROVIDE 24-7 SUPPORT" operation of Fig. 19.

Please replace paragraph [0021] with the following amended paragraph:

[0021] As suggested in Fig. 2A, the tester software 46 and the web browser 48 can be implemented on the same computer apparatus ("machine") or on separate computer apparatus. The case where the tester software 46 and the web browser 48 have been implemented on the same computer apparatus 22', they both can operate under the same ~~operating-system~~ operating system (OS), i.e. OS1=OS2. However, in the instance where the tester software 46 and the web browser 48 are implemented on different computer apparatus, the tester software 46 and the web browser 48 can be supported on different operating systems, i.e. OS1 can be different from OS2. For example, the tester 48 may be implemented under a personal computer operating system such as under Windows® 95,

while the web browser 48 might be implemented on another computer under a workstation operating system such as UNIX®.

Please replace paragraph [0032] with the following amended paragraph:

[0032] As noted in Fig. 4A, the "Add URL Transaction Monitor" refers to a number of steps. This particular web page 96 is created by activating the "SiteScope®" ~~button 96~~ button 97 of the button bar 98. Another button 100 labeled "Reports" can be used to generate test result reports. Therefore, it will be appreciated that the process 46 of Fig. 2B can utilize a web browser window 42 for user interface purposes.

Please replace paragraph [0034] with the following amended paragraph:

[0034] In Fig. 4B, the web page 96 further includes a "disable" box 124 to temporarily disable the monitor sampling and alerting. Next, a step1 POST data field 126 has, as entries, ~~"name = abc" and "address = def"~~ "Name = ABC" and "Address = DEF". The data in the POST data field 126 is used to fill in field parameters. A step1 match content field 128 includes the string ~~"Thank you for entering your name."~~ "your name". A step1 "error if match" field 130 includes the string ~~"invalid entry."~~ "Invalid entry". A step1 authorization user name 132 includes a user ~~name Paul~~ name "PAUL", and step1 password field 134 including a password ~~"secret."~~ "SECRET". Step2 POST data and step3 POST data can likewise be entered.

Please replace paragraph [0035] with the following amended paragraph:

[0035] In Fig. 4C, the end of the step3 POST data and the step4 POST data can be seen on the remainder of ~~page 96~~ web page 96. Below the words "Verify Error" a different portion of the web page 96 is presented which is generic to other parts of the software of the present invention. More particularly, pull down menu 136 permits the update period of minutes, hours, days, etc. to be selected while entry filed 138 indicates the number of seconds, minutes, hours, etc. The update indicates the amount of time between checks whenever the status of the monitor indicates a problem.

Please replace paragraph [0037] with the following amended paragraph:

[0037] Fields 144 and 146 are optional fields ~~which allows~~ which allow a user to enter monitor and report descriptions, respectively. A field 148 permits a time out period in seconds, to wait for the entire sequence to complete. That is, if a monitor session takes more than the time out period, an error or warning will be indicated.

Please replace paragraph [0041] with the following amended paragraph:

[0041] In Fig. 5, a form 158 is displayed to illustrate some of the concepts of the present invention. ~~The form includes~~ form 158 includes a first label 160 "name:" and a second label 162 called "address:". Associated with the labels 160 and 162 are entry fields 164 and 166, respectively. Also associated with the form is a "submit" button 168. The form 158 is as it may appear within a browser window. The HTML which creates the form 158 in the browser window is shown generally at 170. The use of HTML 170 to create a form 158 in a browser window is well known to those skilled in the art.

Please replace paragraph [0042] with the following amended paragraph:

[0042] Fig. 6 illustrates a test configuration file 172 produced by filling out the web page 96 and hitting the "Add" monitor button 122. ~~The test configuration file includes~~ file 172 includes a number of labels or "tags" followed by an equal sign and either a parameter or a blank field. Where the field is blank, the tag is ignored and default values are used, if any.

Please replace paragraph [0043] with the following amended paragraph:

[0043] As noted, the monitor description at 174 is an example of a five step transaction. Other tags such as frequency, time out, error frequency, etc. are filed in, while a number of tags are left blank. For example, since a proxy server is not being used, the tags for the proxy server are all left blank. The tags and parameters for step 1 are shown at 176. The "reference type 1" tag is associated with the parameter "URL". The tag reference 1 is associated with parameter <http://www.freshtech.com>. Steps 2, 3, 4, and 5 are found at 178, 180, 182, and 184, respectively. It should be noted that the step2 ~~type is "blank"~~ type is "link", the step3 type is "form", the step4 type is "URL" and the step5 type is "frame". Therefore, this

example of a test configuration file 172 illustrates four of the main types of features which can be verified and tested by the process of the present invention.

Please replace paragraph [0045] with the following amended paragraph:

[0045] In Fig. 8, a flow diagram illustrates the process 74 of Fig. 2B in greater detail. The process 74 begins at 200, and in an operation 202, the "next step" and cookie is retrieved. The next step is the next test step as described above. The "cookie" is a data file received from the web site being tested that includes information concerning user preferences, user history, etc. An operation 204 determines whether the next step is a null and if so, the process is completed at 206. If there is a next step, ~~operation 208~~ operation 208 determines the step type, as described below.

Please replace paragraph [0047] with the following amended paragraph:

[0047] If operation 208 determines that the step type is a "link", an operation 212 searches the current HTML document for a "link" tag having the specified content. In operation 214 determines whether the appropriate "link" tag has been found and, if not, an error message 216 is ~~generated~~ the generated and the process aborts at 218. If the link tag is found by the operation 214, an operation 220 sends an HTTP request for the URL and the cookie and process control reverts to operation 211.

Please replace paragraph [0048] with the following amended paragraph:

[0048] If operation 208 determines that the step type is "form", an operation 222 searches the current HTML document for a "submit" button with the specified content. An operation 224 determines whether the appropriate "submit" button has been found, and if not, operation 228 determines that there is an error and the process ~~76 aborts at 74~~ 74 aborts at 229. If the appropriate submit button is found by operation 224, an operation 230 finds all fields on the form with the submit button and determines or calculates the default values for this field. An operation 232 then takes the supplied post data and overlays the default values such that all appropriate fields have been filled. An operation 234 then sends an HTTP request "post" plus the cookie and process control ~~returns to 211~~ returns to operation 211.



Please replace paragraph [0049] with the following amended paragraph:

[0049] If operation 208 determines that the step ~~type “frame”~~ type is “frame”, an operation 236 searches the current HTML document for a “frame” tag with specified content. An operation 238 determines whether the appropriate frame tag has been found. If not, an operation 240 indicates an error and the ~~process 76~~ process 74 aborts at 241. If the appropriate frame tag has been found by operation 238, an operation 242 sends an HTTP request for the URL and the cookie. Process control then returns to operation 211.

Please replace paragraph [0050] with the following amended paragraph:

[0050] In Fig. 8A, the operation 230 of Fig. 8 is described in greater detail. The operation 230 begins at 243 and, in an operation 244, all form input elements are determined. If a form input element is not recognized, then the process is complete at 260. If operation 244 finds a radio button input element, operation 248 picks the first radio button for the value. If the operation 244 finds a check-box for the input element, an ~~operation 250~~ operation 250 determines whether the box is checked. If it is, an operation 252 sets the value, and if it is not checked, an operation 254 indicates that there is no value. If an operation 244 finds a text input element, an operation 256 sets the value to the default string. If operation 244 finds pop-up menu, an operation 258 picks the default as the value. The operation 230 is then complete at 260.

Please replace paragraph [0063] with the following amended paragraph:

[0063] In a parsing operation 334, the HTML page is parsed to obtain the next tag. In addition, the HTML page is preferably parsed for image links, such as ~~GIF image~~ Graphics Interchange Format (GIF) image links. Essentially, each HTML page, or form, includes a tag that indicates the start of the form. Further there are various input field tags inside the start tag. Frames include a tag that indicates that the code is for a frame. Generally, all tags and links include a name and a display item that is displayed.

Please replace paragraph [0065] with the following amended paragraph:

[0065] In a branching operation 340, the parsing process 330 branches depending on the tag type obtained in operation 334. The parsing process 330 continues to a get title operation ~~336a-336b~~ 336a-336e depending on the tag type. During the get title operation ~~336a-336b~~ 336a-336e, additional information concerning the tag is obtained.

Please replace paragraph [0066] with the following amended paragraph:

[0066] Preferably, the names and display items are displayed to the user for later selection. Thus, if a link is "click here for food," this display item is extracted from the form and displayed to the user for later selection. Also, input items have names, for example, an input item may have a name such as "First Name" or "Code Number." These names are also extracted. Hence, during the get title operation 336a-336e the descriptive part of the obtained tag is extracted, and added to ~~the to serve~~ serve as categories for user selection in operation 338a-338e. That is, operation 338a adds the descriptive part of the tag to the link category, operation 338b adds the descriptive part of the tag to the form category, operation 338c adds the descriptive part of the tag to the frame category, operation 338d adds the descriptive part of the tag to the input category, and operation 338e adds the descriptive part of the tag to the button category.

Please replace paragraph [0067] with the following amended paragraph:

[0067] ~~Figures 10B-10g~~ Figures 10B-10G illustrate an interface used to create monitors during the process of Figure 10, in accordance with an embodiment of the present invention. More particularly, these are the web pages used to create a "monitor object" which will be used to test and gather data concerning the operation of a web site under test. It should be noted that multiple monitors monitoring multiple web sites or monitoring a single web site in multiple fashions can be operating simultaneously. This is a well-known function of an object oriented programming and of a multi-tasking computer system. However, the following description will be in the context of creating a specific monitor object which will test the functionality of a web page. The user interface may be tailored to the user, either by identifying the user individually, or by association with the user, such as a group or security

level, or some other form of classification. This tailoring may be ~~performed~~ performed before the first test is ~~performed~~ performed, or may be incorporated into the first or latter tests for subsequent use.

Please replace paragraph [0071] with the following amended paragraph:

[0071] As is step2, available links in the step3 page 370 are displayed in the link list 352, available forms and form buttons are displayed in the form button list 354, and available input items are displayed in the an input item list 356. The user is then allowed to select a category to add to the monitor. In the example of Figure 10C, a form "Search" has been selected from the form button list 354. Further a keyword "John" was added (shown in Figure 10D). The user's selection is then added to the monitor in response to a user selection of the "Add Step" button 360. The lists provided can be predetermined by a number of different factors, but in one preferred embodiment information provided on the web page itself will prompt certain selections on the list. Subsequent ~~lists then~~ lists can then also be ~~influnced~~ influenced by a combination of prior selections and information from the web page.

Please replace paragraph [0074] with the following amended paragraph:

[0074] ~~Figure 10e~~ Figure 10E is an illustration showing a step5 monitor editing page 390, in accordance with an embodiment of the present invention. Included in the step5 page 390 is a link list 352, a frame list 364, and new URL text box 357, and function buttons 358, 360, and 362. In the example of ~~Figure 10e~~ Figure 10E, the user has selected a frame "contents" from the frame list 364.

Please replace paragraph [0075] with the following amended paragraph:

[0075] ~~Figure 10f~~ Figure 10F is an illustration showing a step6 monitor editing page 400, in accordance with an embodiment of the present invention. Included in the step6 page 400 is a link list 352, and new URL text box 357, and function buttons 358, 360, and 362. In the example of ~~Figure 10f~~ Figure 10F, the user has decided he is finished editing the monitor, and informs the system by selecting the "Add Monitor" function button 362.

Please replace paragraph [0076] with the following amended paragraph:

[0076] ~~Figure 10g~~ Figure 10G is an illustration showing a completed monitor display 410, in accordance with this embodiment of the present invention. The example of ~~Figure 10g~~ Figure 10G shows that a monitor has been added and successfully run on the selected web page. More particularly, in this example, the monitor took 2.08 sec, and 5 steps to complete. Further shown is the name of the URL transaction and various editing options.

Please replace paragraph [0079] with the following amended paragraph:

[0079] In one embodiment, the data center 452, which is the server location of the site being tested, is in communication with various remote testing locations 454. The testing locations 454 actually test the site at the data center. In communication with the testing locations 454 are customers 456 running browsers 458. Thus, a customer is allowed to test the web site at the data center 452 by controlling a testing location 454 using their browser 458. ~~It should borne~~ should be borne in mind that the various testing locations 454 can use different backbone carriers. For example, one testing location may use Sprint®, while another uses Level 3 Communications®, Inc., and yet another testing location may use MCI® as a backbone carrier.

Please replace paragraph [0083] with the following amended paragraph:

[0083] ~~Figure 13 is a dataflow~~ is a dataflow diagram showing an Enterprise portal system 600, in accordance with another embodiment of the present invention. Included in the Enterprise portal system 600 is a portal 602, and SiteScope® locations 604 in communication with various site servers 606. In addition, each SiteScope® 604 includes its own local database 608, and the portal 602 includes a portal database 610, which is essentially a unified form of the local databases 608.

Please replace paragraph [0084] with the following amended paragraph:

[0084] In use, the portal 602 acts like a collecting point for the information provided by the ~~sitescope~~ SiteScope® locations 604. Specifically, each ~~sitescope~~ SiteScope® location 604 receives monitor results from the various site servers 606 it is monitoring. These monitor

results are then collected by the portal 602. Further, the portal 602 provides monitor configuration information to each ~~siteseer~~ SiteScope® location 604. In this manner, a user 612 may provide monitor configuration information to hundreds of servers merely by providing the monitor configuration information to the portal 602. Once the portal 602 receives the monitor configuration information from the user 612, ~~the portal~~ the portal 602 transmits the monitor configuration information to the appropriate SiteScope® locations 604, which in turn provide the monitor configuration information to specified site servers 606. The user interface is tailored to the user, either by identifying the user individually, or by association with the user, such as a group or security level, or some other form of classification. This tailoring may be performed before the first test is performed, or may be incorporated into the first or latter tests for subsequent use. Tailoring of the interface also included tailoring of information appearing on the interface, including selection choices. This makes the system much easier to use on a per-user basis, and allows for such things as group identification and security clearance.

Please replace paragraph [0093] with the following amended paragraph:

[0093] In the first monitor error operation 662, the particular error achieved is logged, using the database. In one embodiment, the user may further process the error message. A decision is ~~then made~~ then made as to whether to continue the process 650 or to end the process, in operation 661. For every error encountered, the user can define whether the system should continue monitoring or end. In this manner, the user may create complex alert logic for a particular composite monitor. For example, the system could stop monitoring and alert the user when five or more monitors in a group of eight are in error, or when three or more groups have monitors with errors. In addition, the system could be programmed to perform specified actions when these events occur.

Please replace paragraph [0095] with the following amended paragraph:

[0095] In a second pass operation 663, the results obtained in the second monitor operation 660 are compared with predetermined pass criteria. If the results fall within the bounds determined by the predetermined pass criteria, the process 650 continues with the

another monitor operation. However, if the results do not fall within the bounds of the ~~determined~~ determined by the predetermined pass criteria, the process 650 continues with a second monitor error operation 664.

Please replace paragraph [0096] with the following amended paragraph:

[0096] In the second monitor error operation 664, the particular error achieved is logged, using the database. Then, as with the first monitor error operation 662, a decision is ~~then made~~ then made as to whether to continue the process 650 or to end the process, in ~~operation 664~~ operation 661.

Please replace paragraph [0099] with the following amended paragraph:

[0099] As discussed previously, the ebusiness transaction chain monitor 702 monitors a plurality of tested servers, such as the first tested server 704 and the second tested server 708. The ebusiness transaction chain monitor 702 can monitor using the Internet 712. In addition, the ebusiness transaction chain monitor 702 is generally capable of monitoring the tested servers via their backend services, such as the first backend 706 and the second backend 710. One example of a backend service is a relational database. Further, the ebusiness transaction chain monitor 702 may be local to the tested system, or remote as ~~described~~ described with reference to Figure 11.

Please replace paragraph [0115] with the following amended paragraph:

[0115] Figure 19 is a flow diagram illustrating a business ~~method~~ method 1070 for providing a client with a monitoring system, in accordance with an embodiment of the present invention. The process begins with a start operation 1071. In operation 1072, it is determined whether or not the client is or would like to become a Global SiteReliance® Customer. If the client is not a Global SiteReliance® Customer, than the client may purchase stand-alone software in operation 1073. The stand-alone software may include SiteScope® or SiteSeer®. After purchasing the stand-alone software, the client generally has no regularly scheduled contact with the software provider on a long term basis. However, the software

provider may inquire about the stand-alone software's operability and other issues after the initial purchase.

Please replace paragraph [0116] with the following amended paragraph:

[0116] If the client is a Global SiteReliance® Customer, the client's particular monitoring needs are determined in an "interview customer for needs" operation 1074. Upon determining a customer's needs, a customized monitoring system is developed for the client in an "develop a customized solution" operation 1075. The customized monitoring system is developed by configuring the elements illustrated in Fig. 9 to meet ~~the clients~~ the client's needs. This is described in more detail below. In one embodiment, the system may be implemented as a virtual private network between the system provider and the client. This provides the client with added security throughout the monitoring process. Once a customized monitoring system is developed, the system is presented to the client in an "approved by customer" operation 1076. If the monitoring system is not approved by the client, the system is modified to better meet the requirements of the client in an "develop a customized solution" operation 1075. If the customized monitoring system is approved by the client, the system is then implemented.

Please replace paragraph [0120] with the following amended paragraph:

[0120] CentraScope is one embodiment of the portal that gives the client and the software provider a framework for managing large amounts of performance data. As discussed above, the enterprise portal is operable to collect and aggregate data and present it in an organized manner. The presentation may include reports of reliability and other issues related to a monitoring system. It also is able to configure or 'push data' to a large number of SightScope® applications or web servers that CentraScope may interact with. Clients having a large number of servers that require monitoring or configuration would likely ~~implementCentraScope~~ implement CentraScope. However, CentraScope is merely an optional component of the monitoring system.

Please replace paragraph [0121] with the following amended paragraph:

[0121] SiteSeer® is an application used for instrumenting the monitoring from locations around the world. In one embodiment, the monitoring is done from popular business centers around the world in countries such as England, France, Germany, Australia, United States, Japan, Canada, and other countries that a client may expect their respective clients to access their web services from. SiteSeer® may perform numerous transactions ~~with a clients~~ with a client's website, monitor and record the results for the transactions, record the performance data, and then process the resulting information. One advantage of this system is that monitoring from different countries will render different results and response times. This enables a client to determine weaknesses in their systems as viewed from different locations around the world. SiteSeer® may also set alerts for web site response time and other elements of the client's web site networking. For example, a client may want to be alerted if a response time for their web site exceeds seven seconds. If an alert is received, SiteScope® and the software companies supporting staff may determine the cause of the alert from the SiteScope® reports of the monitoring. The reports may indicate that the client hardware or software needs to be updated. In some instances, a client may solve an alert regarding a slow response time in another country by caching information at different locations. The reports may also indicate that a networking path or a service provider is down, thereby informing a client of the reliability of their service provider and of a possible need to consider back-up systems or a change in their provider. Further, the monitor systems implemented on SiteScope® at the client location may be the same monitors implemented on the SiteSeer® remote nodes. This allows for a client to compare monitoring issues from internal and external points of view, thus suggesting a firewall problem or problem somewhere else within the perimeter of the system. Clients that provide web resources and services to their respective clients in different locations and countries would likely implement the SiteScope® application within their monitoring system.

Please replace paragraph [0122] with the following amended paragraph:

[0122] The Best Practices Library is a text-based guide that addresses common failure or stress points for different web based systems. In one embodiment, the library is available to



the client through a web site that the client may access over the Internet. The library is a broad document that is then customized for each client depending upon their specific web-based applications and services. Different clients have different needs for their respective clients. For example, some clients such as internet service providers provide web access according to a service level agreement while other clients simply represent to their respective clients that their web site is available to available at all times. As the clients monitoring system is configured and developed, their respective library is updated to better address the clients needs regarding their system. This involves eliminating information not relevant to the particular client as well as making general information more specific to the client's needs. A portion of the Best Practices application is applicable to several types of services, such as monitoring routers and queues, monitor the bottlenecks and keep information moving forward, queues that monitor outgoing information against incoming information such as sending confirmation information, and other recognizable problem spots. All clients are provided with a Best Practices Library which is updated as the clients monitoring system is developed and enhanced.